International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I9P116
Volume 72 Issue 9, 197-202, September 2024

Review Article

Evaluating the Real-World Application Efficacy of MobileNet Models

Sara Bouraya[1], Abdessamad Belangour[2]

[1,2]*Laboratory of Information Technology and Modeling, Hassan II University, Faculty of Sciences Ben M'sik Casablanca, Morocco.*

[1]*Corresponding Author : sarabouraya95@gmail.com*

Received: 22 May 2024 Revised: 08 August 2024 Accepted: 02 September 2024 Published: 28 September 2024

Abstract *- This experimental study explores the abilities of MobileNet and its three variants within the sphere of object classification for object detection under different lighting. Our research trains every model on the 'Car Object Detection' dataset with adjustments to lighting, weather conditions, and urban or rural settings, which represent real-life situations more accurately. We outline specific alterations made to architecture and methods used during training that were meant to increase adaptability across different environments while maintaining accuracy, too. As a result, this work achieved remarkable results, and our best-performing algorithm attained a 97% validation accuracy rating according to tests carried out under various environmental conditions. Through lightweight convolutional networks for object detection, it becomes clear that such type was not only effective but also resource efficient, hence applicable in dynamic settings requiring real-time operation with limited resources.*

Keywords *- CNNs, Convolutional Neural Networks, Computer Vision, MobileNet, Object Classification.*

1. Introduction

Computer vision relies greatly on object detection, which is crucial in the development of autonomous driving, security surveillance systems and traffic management, among others. Many factors still challenge the practicality and trustworthiness of real-life object detection models. This article probes into how effective these deep learning models handle complex situations with a special emphasis on transfer learning techniques. In most cases, object detection models can be categorized as either one-stage or two-stage detectors. For quickness purposes, one-stage detectors like YOLO [1], SSD [2], RetinaNet, YOLOv4 [3], YOLOv2 [4], CornerNet [5], Scaled-YOLOv4 [6], CenterNet [7] and ThunderNet [8] were designed. These detect classes of objects together with their locations in just one step without having to generate region proposals first. Conversely, two-stage detectors such as R-CNN [9], Fast R-CNN [10], Faster R-CNN [11], Mask R-CNN [12] and Cascade R-CNN [13] create region proposals before classifying each one into different object categories since they are more concerned about accuracy than speed. Each type of model has its own upsides for real-time processing applications. Our team has investigated a range of computer vision components in previous research. For example, we have looked at object tracking, object detection, and the examination of different neck models in object detection systems. These studies were fundamental in that they provided necessary performance measures and implementation aspects for different methods and frameworks used. Another notable achievement from our earlier work was the production of a paper that classified video datasets systematically, thus creating an important reference material for researchers in this area.

This paper is the end result of a lot of preliminary work that has been done in computer vision research. It uses based studies that have looked at different parts of object detection, such as tracking methods and model structures. Here, we move from just thinking about things to actually doing them by running tests on how well these models can detect objects in real-life situations. What this research does is try to connect theory with practice.

2. Related Work

The area of research into efficient deep learning architectures for mobiles and edge computing has been very much alive, with MobileNet models leading because they are the most efficient. This part examines the history of MobileNet structures and their assessment in diverse applications, offering a base for our experimental study on different forms of MobileNet. In their work, Howard et al. proposed an original MobileNet model that employs depthwise separable convolutions to reduce computation cost and model size significantly, hence making it suitable for mobile devices[14]. This design was seminal as it showed how Convolutional Neural Networks (CNNs) can be optimized for performance on hardware with limited resources. MobileNetV2 was developed by Sandler et al., which built on the backbones of the initial one and introduced inverted residuals and linear bottlenecks. This version sought to improve efficiency through optimizing layer structures and intra-network data flow further enhancing its performance in mobile devices [15]. MobileNetV3 then came along, and Howard et al. employed AutoML together with network pruning techniques to optimize it even more.

Hardware-aware network design was combined with novel architectural strategies like using squeeze-and-excitation blocks within MobileNetV3 so as to increase both processing speed and accuracy [16]. There have been many comparisons made between how well different models perform when trying to achieve efficiency, such as ShuffleNet or EfficientNet against MobileNets themselves. An example of such comparison is ShuffleNet V2 by Ma et al., who took direct benchmarks against MobileNetV2, thus showing its superior speed advantages over similar computational constraints in terms of model size [17].

3. Background

The MobileNet family of models is a significant improvement in efficient deep learning architectures designed mainly for mobile and edge devices with limited computational resources and power consumption. Here is an overview of the MobileNet models, including MobileNet, MobileNetV2, MobileNetV3 Small, and MobileNetV3 Large.

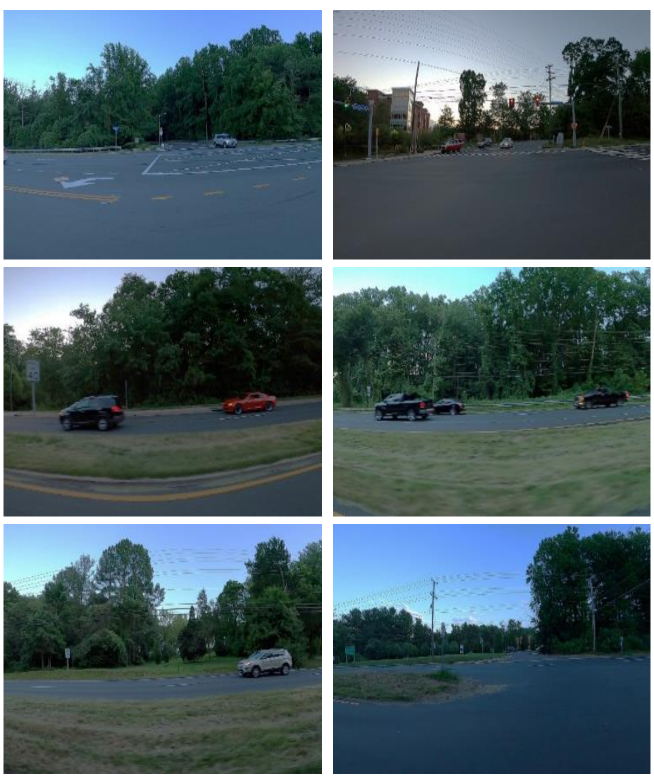


Fig. 1 An overview of the car object detection database

3.1. MobileNet

In 2017, Howard et al. created the original MobileNet model, which introduced depthwise separable convolutions. The convolution process is split into two layers through this method: a depthwise convolution that applies one filter per input channel and a pointwise convolution that uses 1x1 convolution to combine outputs from the depthwise layer. This technique significantly reduces computational cost and parameter count, thereby making it highly efficient without sacrificing accuracy. Also, it can be easily adapted for various image recognition tasks across different domains [14].

3.2. MobileNetV2

Sandler et al. released MobileNetV2 in 2018 building upon the success achieved with MobileNets. In this version, two main novelties are brought to light: inverted residuals and linear bottlenecks. Lightweight depthwise convolutions work as residual learning filters using inverted residual structures. At the same time, a linear bottleneck controls the flow of features through the network, thus enhancing the efficiency and effectiveness with which the model operates, especially when dealing with non-linearities at low-dimensional representations [15].

3.3. MobileNetV3 Small & Large

AutoML, along with network pruning techniques, were used to optimize the MobileNetV3 models developed by Howard et al. in 2019. Two versions of this model were provided, namely, a small version for devices having lower power budgets but still efficient yet accurate enough models and a large variant which strikes a balance between efficiency and accuracy suitable for slightly higher capacity devices like smartphones or tablets, etc. There are some advanced features included within these architectures, such as squeeze-and-excitation blocks, that improve the network's representational capacity, among others. Also, hard-swish combined with other non-linearities were used specifically tailored for low-power devices [16].

4. Methodology

4.1. DataSet

The dataset for this study comprises one thousand one hundred seventy-six (1,176) images. It is split into two groups: training and testing images. This division is done in order to evaluate the object detection models more effectively. The number of pictures in the training subset is exactly one thousand and one (1,001). Each image was handpicked and labeled accordingly, and it represents different scenarios commonly faced by vehicles. Various types of vehicles are shown in those pictures, including small cars, big trucks or vans, etcetera, while they are being taken in different environmental settings. For example, some were captured during a bright sunny day, whereas others were photographed under low light conditions like dawn or dusk; still, many more were shot in complete darkness as it happens during nighttime hours. The selection also covers objects viewed through raindrops or mist, which could cause difficulty for the detectors when trying to identify them correctly if need be considered. Finally, there were shots taken from busy city streets full of people up to deserted country roads with no living soul around except for maybe an occasional animal crossing the path somewhere deep inside a forest, which could make any algorithm struggle even harder because such backgrounds pose additional challenges related not only to recognition but also tracking moving targets against static ones amidst clutter like trees' branches interlaced together. The testing set is also diverse, although it has only 175 images, and it evaluates how well the trained models generalize. This subset consists of a variety of challenging conditions like those found in the training set but does not have any image duplicates or exact scenarios. Such a method gives a good idea about how good our model can work with different data sets. The training and test set both include environmental challenges that help assess and enhance the robustness of detection algorithms by making them perform accurately under less favorable visual conditions.

4.1.1. Models Architecture

Figure 2 shows a neural network architecture with MobileNet as the key element. It starts with an Input Layer, which processes image data designed for images of size 224×224 pixels with 3 color channels (RGB). This is common in most image recognition tasks as it allows the model to work on standard input sizes for real-world applications. A MobileNet Functional Layer, referred to as "mobilenet_1.00_224", follows the Input Layer in this design. It is responsible for doing most of the computation such as extracting features from an image. Its output is a reduced feature map with dimensions of 7×7 and 1024 channels, i.e., compressing the image data into a form that can be easily managed while still keeping vital information intact. A Global Average Pooling 2D layer then takes in outputs from the MobileNet layer. What this does is that it simplifies things by taking the average across all entries at each of its 1024 channels, thereby collapsing them to a one-dimensional array having 1024 elements only. Global Average Pooling tremendously helps reduce model complexity and prevents overfitting by minimizing the number of trainable parameters. Further protection against overfitting comes through the use of Dropout layers employed as a means of regularization in this network topology. These layers drop out input units randomly during training, i.e., they set some fractions them equal to zero, which helps make the network more robust towards noise, and different forms of input data may present themselves. Finally, there is a dense layer that produces the final predictions. From what I can tell, it seems like a binary classification task since the dense layer shown here outputs a single value indicating either positive or negative class membership probability but not both simultaneously. Such a setup would work best where the highest computational efficiency is desired, such as mobile or edge computing, given the tradeoff between speed and cost involved here.

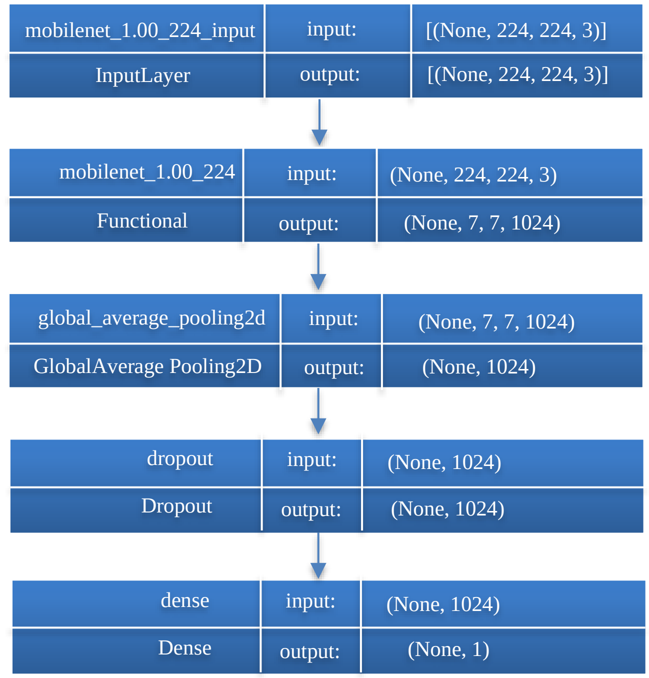


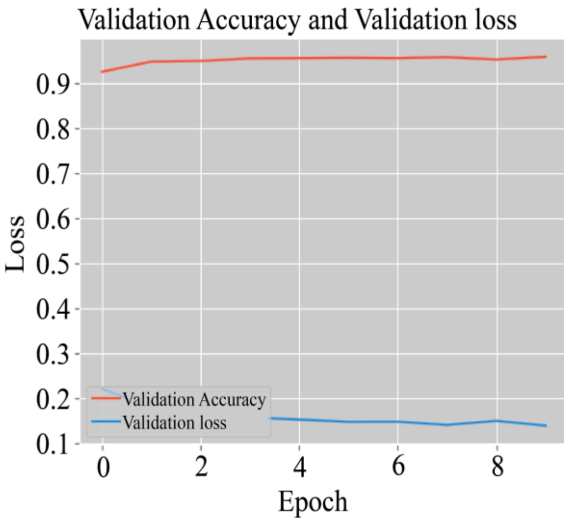
Fig. 2 Enhancing Image Classification with MobileNet: A Transfer Learning Approach

5. Results

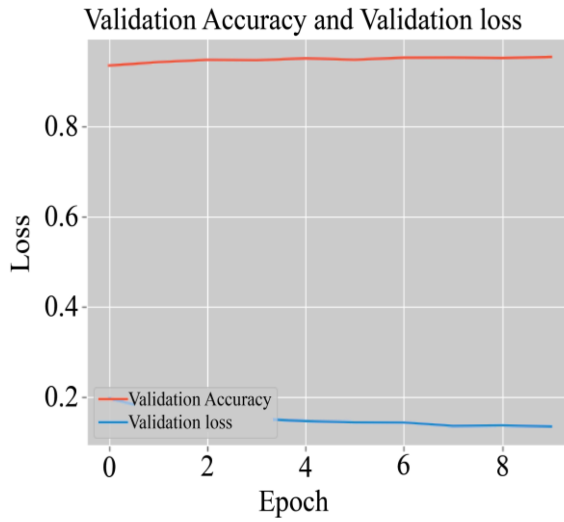
Table 1 presents the performance metrics for four different models, each based on a variant of the MobileNet architecture, highlighting their training and validation accuracies. Model 1, built on the original MobileNet architecture, achieved a training accuracy of 93.68% and a validation accuracy of 95.79%. This model showcases the balance between efficiency and performance that MobileNet aims to provide, especially for mobile environments. Model 2 utilizes the MobileNetV2 architecture, which introduces inverted residuals and linear bottlenecks to enhance processing efficiency. It registered a training accuracy of 94.68% and a validation accuracy of 95.50%, slightly lower than Model 1, suggesting variations in how each model handles overfitting and generalizes to new data. Moving to the more advanced MobileNetV3 architectures, Model 3 employs the MobileNetV3 Small variant, optimized further for performance with techniques like AutoML and network pruning. This model outperforms the earlier versions with a training accuracy of 96.85% and a validation accuracy of 97.22%, demonstrating improved efficiency and capability in handling complex tasks on power-constrained devices.

Table 1. Performance Metrics of MobileNet-Based Models in Object Detection

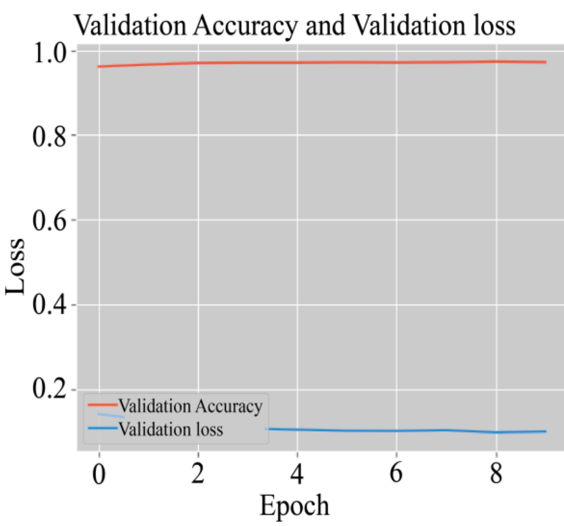
Model	Base Model	Training Accuracy	Validation Accuracy
Model 1	MobileNet	0.9368	0.9579
Model 2	MobileNetV2	0.9468	0.9550
Model 3	MobileNetV3Small	0.9685	0.9722
Model 4	MobileNetV3Large	0.9753	0.9791



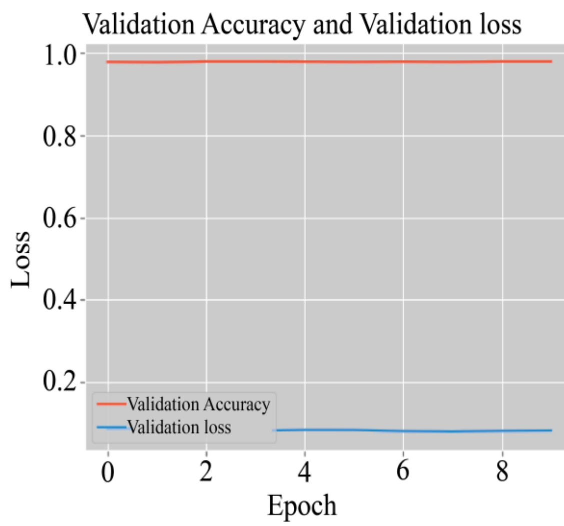
Model Based MobileNet



Model Based MobileNetV2



Model-Based MobileNetV3Small



Model-Based MobileNetV3Large

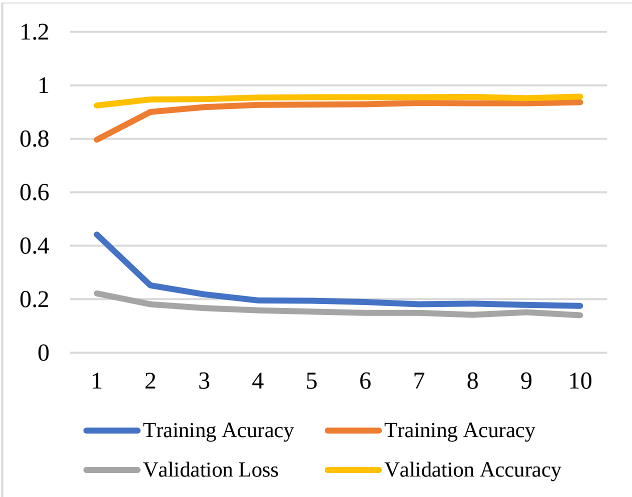
Fig. 3 Evaluation metrics of several MobileNet variants: Training accuracy and training loss

Model 4, based on MobileNetV3 Large, achieves the highest accuracies among the group, with a training accuracy of 97.53% and a validation accuracy of 97.91%. This version is designed to balance computational efficiency with higher accuracy, making it ideal for more demanding applications that require precise image recognition capabilities (Figure 3).

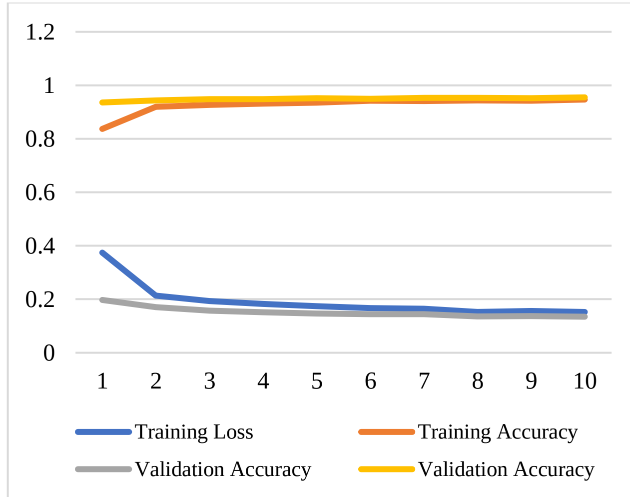
6. Discussion

MobileNet and MobileNetV2 exhibit what looks like a rapid decline in training loss, and their validation loss also drops progressively. This shows that they have good generalization without overfitting the data too much. However, MobileNet has slightly more validation loss than MobileNetV2, which implies some slight improvements may have been made to it for better optimization of models as well as improving its ability to generalize those models. MobileNetV3 Small demonstrates a remarkable convergence with a steep descent in training loss and consistently low validation losses that are among the least achieved across all the versions considered; this suggests very effective learning combined with generalization, possibly due to advanced architectural features and optimizations. MobileNetV3 Large performs equally well where it achieves the lowest validation loss together with the highest validation accuracy, indicating great design modifications, which makes it the most robust model for complex object detection tasks, among others.

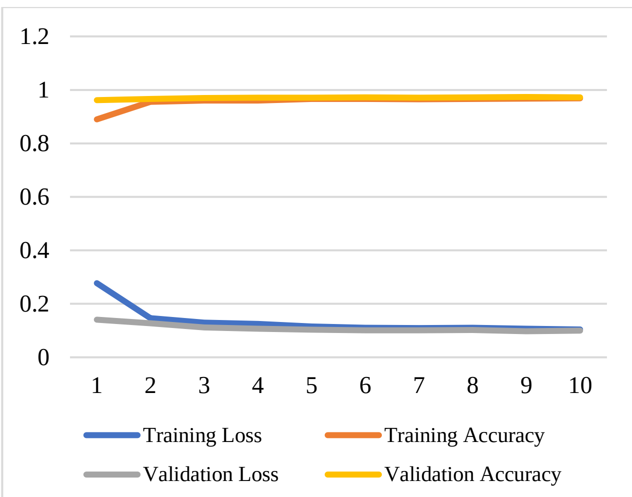
Training accuracy increases steadily in all cases until MobileNetV3 Large achieves near-perfect levels of accuracy. Similarly, MobileNetV3 Large tops in terms of validation accuracy while MobileNetV3 Small comes second close behind it; however, older models, though effective, do not reach such high values of accuracy, showing how much better V3 iterations have become (Figure 4).



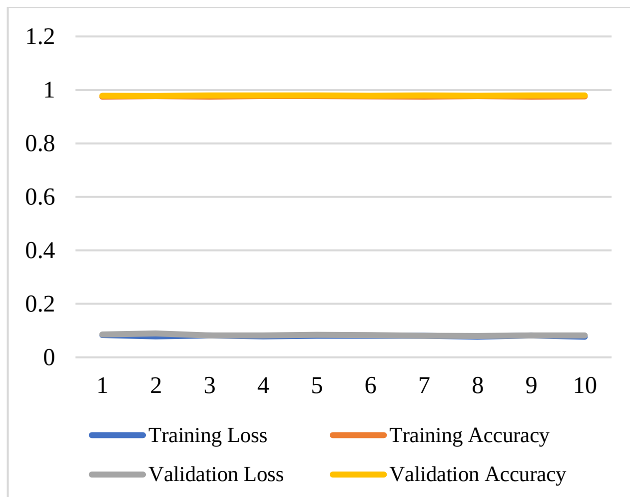
Model Based MobileNet



Model Based MobileNetV2



Model-Based MobileNetV3Small



Model-Based MobileNetV3Large

Fig. 4 Evaluation Metrics of Several MobileNet Variants: Validation Accuracy and Validation Loss

7. Conclusion

According to our research, every one of the four versions of Mobilenet achieved very high accuracy rates, which demonstrates its ability to handle complex tasks in object detection efficiently, among other things.

The best model outperformed others by achieving the highest accuracy score and being the most stable across various metrics used for measurement, such as mobile net large v3. It had better features based on architecture as well as optimizations that greatly contributed towards its improved performance over others, thus making it ideal when there is a need for higher precision at low computational cost especially where accuracy matters most. These incremental changes between mobilenet through mobilenetv3 large show continuous development in network design and optimization techniques that not only improve performance but also help models to learn from training data so that they can work well even beyond their training environment, hence becoming more useful for different practical purposes.

References

[1] Joseph Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 779-788, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Wei Liu et al., “SSD: Single Shot Multibox Detector,” *Computer Vision – ECCV: 14th European Conference*, Amsterdam, The Netherlands, pp. 21-37, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv*, pp. 1-17, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Joseph Redmon, and Ali Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 6517-6525, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Hei Law, and Jia Deng, "CornerNet: Detecting Objects as Paired Keypoints," *International Journal of Computer Vision*, vol. 128, pp. 642-656, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 13024-13033, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kaiwen Duan et al., "CenterNet: Keypoint Triplets for Object Detection," *2019 IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), pp. 6568-6577, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Zheng Qin et al., "ThunderNet: Towards Real-Time Generic Object Detection on Mobile Devices," *2019 IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), pp. 6717-6726, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ross Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580-587, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ross Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision*, Santiago, Chile, pp. 1440-1448, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[11] Shaoqing Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Kaiming He et al., "Mask R-CNN," *2017 IEEE International Conference on Computer Vision*, Venice, Italy, pp. 2980-2988, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Zhaowei Cai, and Nuno Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 6154-6162, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] Andrew G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv*, pp. 1-9, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mark Sandler et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4510-4520, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Andrew Howard et al., "Searching for MobileNetV3," *2019 IEEE/CVF International Conference on Computer Vision*, Seoul, Korea (South), pp. 1314-1324, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ningning Ma et al., "ShuffleNet V2 : Practical Guidelines for Efficient CNN Architecture Design," *Computer Vision – ECCV: 15th European Conference*, Munich, Germany, pp. 122-138, 2018. [CrossRef] [Google Scholar] [Publisher Link]